Atari Computer Enthusiasts (N.S.W.)

INSIDE INFO No. 7 June 1983

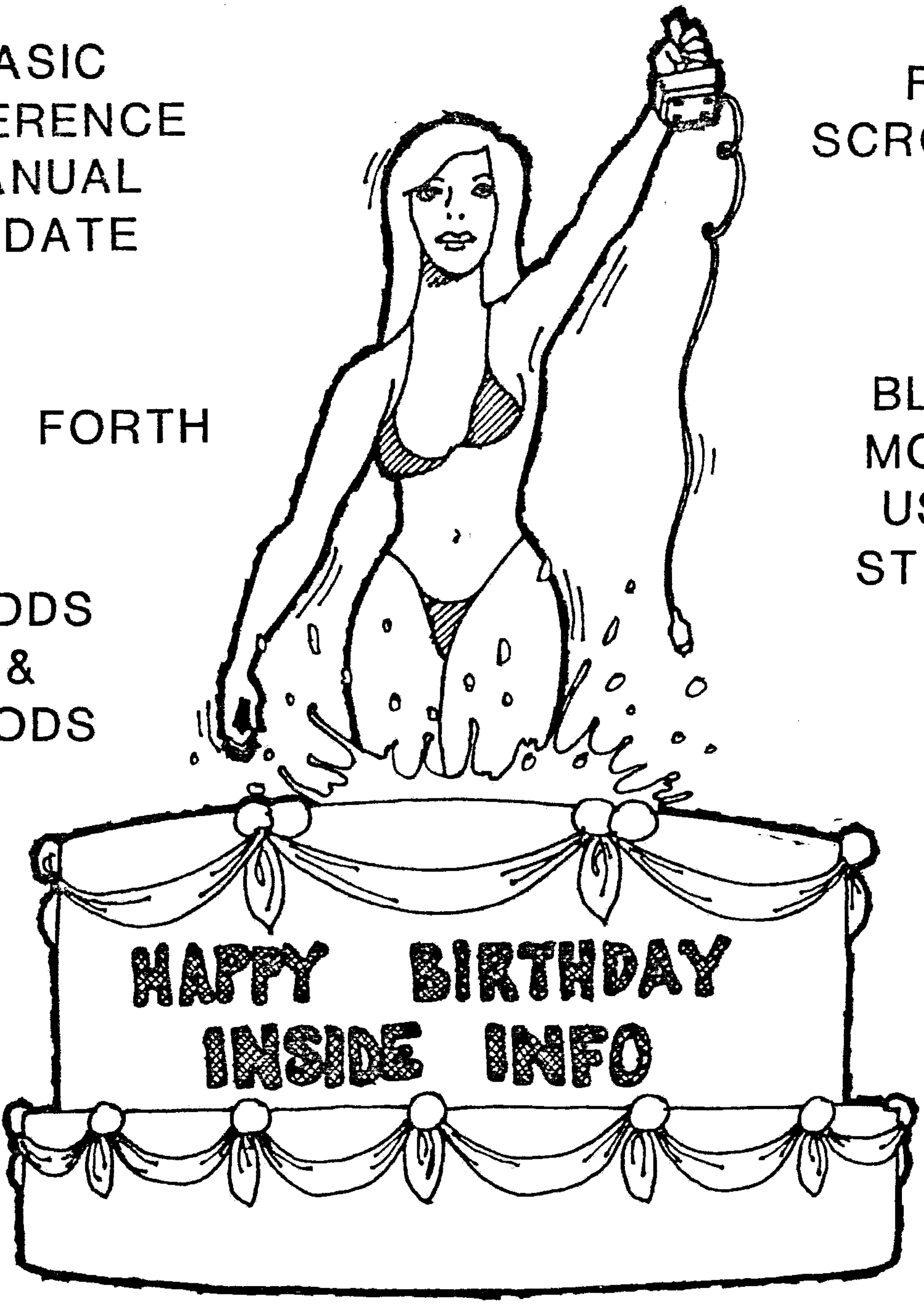
BASIC REFERENCE MANUAL UPDATE

FORTH

ODDS & SODS

FINE SCROLLING

BLOCK MOVES USING STRINGS



Atari Computer Enthusiasts (N.S.W.)

Atari Computer Enthusiasts (N.S.W.) is an independent, non-profit computer users' group loosely affiliated with Atari Computer Enthusiasts in the U.S.A. We have no connections with ATARI, Inc. or their Australian distributors, Futuretronics Australia Pty Ltd. Our aims include promotion of the ATARI 400/800 Home Computer System, instructing both beginners and advanced users in programming techniques, exchanging hints, tips and ideas amongst members and generally enjoying ourselves.

Meetings are held at 6.0 P.M. on the first Monday of every month (or the second Monday if it clashes with a public holiday) in the Amenities Room, 7th Floor, OTC House, 32-36 Martin Place, Sydney. Sign the attendance book in the foyer on your way in, take the lift to the 7th floor and turn right when you get out of the lift. Visitors welcome.

Membership to A.C.E.(N.S.W.) is $30 for the first year and $15 per year thereafter (or $20 and $10 respectively for students under 18 and still at school). Subscriptions, postal enquiries and all correspondence should be directed to:

Atari Computer Enthusiasts (N.S.W.)
G.P.O. Box 4514,
Sydney,
N.S.W. 2001

Phone contacts:

Tony Reeve (President): 452-2974 (Home) or 923-9586 (Work)
Barry Williams (Vice President): 452-2229 (Home) or 359-2424 (Work)
Steven Marcus (Secretary/Treasurer): 387-4287 (Home)
John Massara (Software Exchange): 713-6798 (Home) or 428-2866 (Work)
Peter Bamford (Meetings Organiser): (043) 42-2655 (Home) or (043) 40-1201 (Work)

Meeting Dates

*6th June
4th July
*1st August
5th September
*10th October (due to 8-Hour Day Public Holiday)
7th November
*5th December

* indicates release dates for INSIDE INFO (subject to unforseen circumstances). Deadline for articles is the meeting prior to the release date.

New Members

A. Alberth, Rubyvale, Qld.
G. Ashworth, Mt. Riverview
Dale Baikie, Ryde
R. T. Dixson, Macquarie Fields
N. Edwards, Bundaberg, Qld.
J. Ellard, Balmoral Beach
John Fairburn, Para Hills, S.A.
S. Fish, South Strathfield
T. Garkawe, Toorak, Vic.
G. Gough, Bundaberg, Qld.
Peter Gray, Kotara
David Harrington, Canberra, A.C.T.
David Hutt, Allambie Heights
M. Johnson, Goulburn
E. Kindynis, Surry Hills
J. W. Latham, Bundaberg, Qld.
C. Van Oosterom, O'Connor, A.C.T.
N. V. Pearce, Norwood, S.A.
D. R. Pepperell, N.T.
M. Picthall, Bellevue Hill
Sean Quinlan, St. Ives
R. Reynoldson, Cootamundra
G. Rhall, Gladesville
Michael Richardson, Lithgow
Craig Rodger, Ballina
C. Spence, Terrey Hills
R. Taylor, Hebersham
R. C. Taylor, Wahroonga
Geoffrey Thom, Lugarno
C. Thurmer, Epping

Editorial

This is the first anniversary of INSIDE INFO. I thought we'd never make it! As promised, this issue has 2 articles on fine scrolling - advanced programmers will find these particularly interesting. There's also a very useful technique for doing fast block moves using strings and an introduction to the Team ATARI version of fig-FORTH available from the Software Exchange for only $10. The main feature is a 6 page supplement of revisions for the BASIC Reference Manual. I believe this is now included with the BASIC cartridge, but that's not much good to you if you bought BASIC some time ago. We thought it was in all members' best interests to be made aware of these revisions, as they warn of all the bugs in the BASIC cartridge (surprisingly few compared to other computers) and the misprints in the original manual. The Thorn EMI reviews have been held over 'til a later date, as only two of the reviewers have come forward with the goods.

I had a short burst of article submissions about 3 months ago and haven't received any since! The inwards tray is now empty. We want articles! There will be no issue in August unless I am inundated with fresh material. You've been warned!

Looking for ideas? Sometime in the dim, distant future, I'd like to do a special issue on Adventures and another on simple hardware projects. Can anybody help with these? I especially need help with the hardware stuff, as I'm a software expert only. Hardware is foreign ground. I've got a couple of articles from overseas users' groups, but I need someone to check them out. This entails checking the diagrams, converting for Australian conditions, searching for parts, possibly building the projects and all that sort of stuff. Please help!

You may have noticed that articles are tending towards more and more advanced applications. In future, I'd like to offset this by having a beginners' page every issue. Can you help?

Did you get the relationship between the cartoons on the cover of the last issue and the contents within? I want a cartoonist to do similar covers in the future. I can't devote the time to it anymore, so can someone please come to the rescue? Send a sample of your work to The Editor.

It has become increasingly necessary to include an Information Page in every issue. From next issue on (if there is one), you'll be able to get information on back issues, blank media, Software Exchange, bulk purchases, Special Interest Groups, meeting dates, membership details, advertising rates, etc., etc., etc. Please ensure that all info for that page is submitted as soon as possible. Committee members take note!

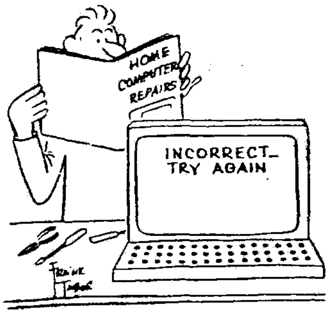
And now the news you've all been waiting for. The winner of the 1st Annual Award for the Most Popular Author goes to some bloke by the name of Garry Francis (whoever he is). He came in with a whopping 68% of the vote. I expected votes to be heavily weighted towards articles from the last 2 issues, as new members don't have earlier issues. Quite surprisingly, the single most popular article, with 20% of the vote, was "GTIA" by Garry Francis. This was in Issue No.3. Other popular articles were "Anatomy of Doggies" by Garry Francis (Issue No.5), "Grab An Apple" by Jamie Athas and Garry Francis (Issue No.6), "Squares" by John Massara (Issue No.4) and "Odds & Sods" by Garry Francis (every issue except No.3). The other authors who received votes were Jamie Athas and Gregg Ramsey. Thanks to all those who voted (did you vote?) and especially to those who voted for me. It makes the blood, sweat and tears of the past year seem just that little bit more bearable.

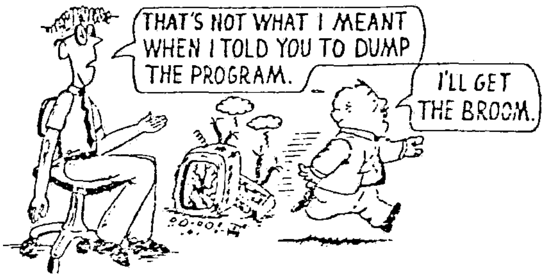
Finally, I am no longer available by 'phone. Do not try to ring me at the old number under any circumstances, as you will only disturb my parents! All correspondence should be addressed to The Editor, A.C.E.(N.S.W.), G.P.O. Box 4514, Sydney, N.S.W., 2001.

- Garry Francis

```
1 REM ‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡
2 REM ‡  HORIZONTAL SCROLLING DEMO  ‡
3 REM ‡       by J.S. Masters       ‡
4 REM ‡  Modified by Garry Francis  ‡
5 REM ‡ Published by Atari Computer ‡
6 REM ‡    Enthusiasts (N.S.W.)     ‡
7 REM ‡          June 1983          ‡
8 REM ‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡
10 GOSUB 1000
20 GOTO 20
999 REM *** Initialisation ***
1000 GRAPHICS 8:POKE 559,0:POKE 16,0:P
OKE 53774,0:RAMTOP=PEEK(106):POKE 708,
200:POKE 710,40
1009 REM *** Redefine Characters ***
1010 START=RAMTOP-8:CH=256*START:FOR I
=CH TO CH+47:READ A:POKE I,A:NEXT I:PO
KE 756,START
1020 DATA 0,0,0,0,0,0,0,0,1,3,7,15,31,
63,127,255,128,192,224,240,248,252,254
,255,255,255,255,255,255,255,255,255
1030 DATA 0,16,56,124,84,84,130,254,16
,16,56,56,84,170,84,170
1039 REM *** New Display List ***
1040 TEMP=RAMTOP-20:DM=256*TEMP:FOR I=
1536 TO 1538:POKE I,112:NEXT I
1050 FOR I=1539 TO 1572 STEP 3:POKE I,
87:POKE I+1,0:POKE I+2,TEMP:TEMP=TEMP+
1:NEXT I
1060 POKE 1575,65:POKE 1576,0:POKE 157
7,6:POKE 560,0:POKE 561,6
1069 REM *** P-M Graphics ***
1070 POKE 54279,START:PM=256*START+102
4:FOR I=PM+70 TO PM+76:READ A:POKE I,A
:NEXT I
1080 POKE 53256,0:POKE 704,56:POKE 532
48,90:POKE 623,4:POKE 559,58:POKE 5327
7,2
1090 DATA 224,124,254,255,254,124,224
1099 REM *** Draw Display ***
1100 BOTTOM=DM+2816:FOR X=0 TO 255:SOU
ND 0,X,10,8:READ Y,CHAR:OFFSET=DM+256*
Y+X:POKE OFFSET,CHAR
1110 FOR I=OFFSET+256 TO BOTTOM+X STEP
 256:POKE I,3:NEXT I:NEXT X:SOUND 0,0,
0,0
1120 DATA 9,132,9,1,8,1,7,133,8,2,8,1,
7,1,6,1,6,2,6,197,7,132,8,196,8,1,7,1,
7,2,7,1,6,1,5,197,6,2,7,2
1130 DATA 8,2,9,2,9,133,9,1,8,1,8,2,9,
2,9,1,8,1,7,132,7,1,7,2,7,1,6,1,5,133,
5,1,4,1,4,2,5,2,5,1
1140 DATA 4,1,3,1,2,1,2,2,2,1,2,2,3,2,
3,1,2,1,2,2,3,2,3,197,4,196,5,2,6,2,6,
132,6,1,6,2,6,1,5,197
1150 DATA 5,1,4,133,4,1,4,2,4,1,4,2,5,
2,6,2,7,2,8,2,9,2,9,196,9,1,9,2,9,196,
10,2,10,1,9,1,8,1,7,1
1160 DATA 7,2,8,2,8,1,7,1,6,1,5,1,5,2,
6,2,6,1,6,2,6,1,5,197,5,1,4,196,4,1,4,
2,4,197,5,2,6,2,6,1,5,1
1170 DATA 4,1,3,1,3,2,3,1,2,1,2,2,2,1,
2,2,3,2,3,1,3,2,4,2,5,2,6,2,6,133,6,1,
6,2,6,1,6,2,7,2
1180 DATA 8,2,8,132,9,2,9,1,9,2,9,1,9,
2,9,1,8,1,7,1,6,196,7,2,7,133,7,1,6,1,
5,1,4,1,3,1,2,1,1,1
1190 DATA 1,2,2,2,3,2,4,2,4,196,5,2,5,
132,6,2,7,2,8,2,9,2,9,1,9,2,10,2,10,1,
10,2,10,1,9,133,9,1,9,2
1200 DATA 10,2,10,197,10,1,9,1,8,1,7,1
,6,132,6,1,5,197,5,1,4,1,3,1,2,1,2,2,3
,2,4,2,5,2,6,2,7,2,7,1
1210 DATA 6,1,5,1,4,1,4,2,5,2,6,2,7,2,
8,2,9,2,10,2,10,1,10,2,10,1,9,1,8,1,8,
2,8,197,8,1,7,1,6,1
1220 DATA 5,1,5,2,6,2,6,1,5,1,5,2,6,2,
7,2,8,2,8,1,7,1,6,1,6,2,7,2,7,196,8,2,
9,2,9,196,9,132,9,1
1230 DATA 8,1,7,1,7,2,8,2,8,1,7,1,6,1,
5,1,5,2,6,2,7,2,8,2,9,2,9,132,9,1,8,1,
7,133,8,2
1240 DATA 8,1,7,1,6,1,6,2,6,197,7,132,
8,196,8,1,7,1,7,2,7,1,6,1,5,197,6,2,7,
2,8,2,9,2
1249 REM *** VBI Routine ***
1250 FOR I=1578 TO 1637:READ A:POKE I,
A:NEXT I:X=USR(1578)
1260 DATA 104,169,0,133,203,141,4,212,
160,59,162,6,169,7,76,92
1270 DATA 228,198,203,165,203,141,4,21
2,16,31,169,7,133,203
1280 DATA 141,4,212,238,4,6,173,4,6,20
1,234,208,2,169,0,162
1290 DATA 0,157,4,6,232,232,232,224,39
,208,246,76,98,228
1300 RETURN
```

```
            0100 ;##############################
            0110 ;# MACHINE LANGUAGE STUFF FOR  #
            0120 ;#  HORIZONTAL SCROLLING DEMO  #
            0130 ;#      by Garry Francis       #
            0140 ;# Published by Atari Computer #
            0150 ;#   Enthusiasts (N.S.W.)      #
            0160 ;#          June 1983          #
            0170 ;##############################
            0180 ;
00CB        0190 SHADOW =    $CB
0604        0200 LMSLOW =    $0604
E45C        0210 SETVBV =    $E45C
E462        0220 XITVBV =    $E462
D404        0230 HSCROL =    $D404
            0240 ;
            0250 ;ROUTINE TO ZERO HSCROL AND SET VBI VECTOR
            0260 ;
0000        0270        *=   $062A
062A 68     0280        PLA             ;Pop no. of arguments off stack
062B A900   0290        LDA  #$00       ;Zero HSCROL (and shadow register)
062D 85CB   0300        STA  SHADOW
062F 8D04D4 0310        STA  HSCROL
0632 A03B   0320        LDY  #MYVBI&255 ;LSB of VBI Service Routine
0634 A206   0330        LDX  #MYVBI/256 ;MSB of VBI Service Routine
0636 A907   0340        LDA  #$07       ;Indicates deferred VBI
0638 4C5CE4 0350        JMP  SETVBV     ;Set vector
            0360 ;
            0370 ;VERTICAL BLANK INTERRUPT SERVICE ROUTINE
            0380 ;
063B C6CB   0390 MYVBI  DEC  SHADOW     ;Decrement HSCROL
063D A5CB   0400        LDA  SHADOW
063F 8D04D4 0410        STA  HSCROL
0642 101F   0420        BPL  END        ;Reached zero?
0644 A907   0430        LDA  #$07       ;Yes...reset to maximum
0646 85CB   0440        STA  SHADOW
0648 8D04D4 0450        STA  HSCROL
064B EE0406 0460        INC  LMSLOW     ;Increment LMS low byte
064E AD0406 0470        LDA  LMSLOW
0651 C9EB   0480        CMP  #$EB       ;Exceeded maximum?
0653 D002   0490        BNE  SKIP
0655 A900   0500        LDA  #$00       ;Yes...reset to zero
0657 A200   0510 SKIP   LDX  #$00       ;Zero counter
0659 9D0406 0520 LOOP   STA  LMSLOW,X   ;Adjust LMS for each mode line
065C E8     0530        INX             ;Increment counter by 3
065D E8     0540        INX
065E E8     0550        INX
065F E027   0560        CPX  #$27       ;Counter exceeded maximum?
0661 D0F6   0570        BNE  LOOP
0663 4C62E4 0580 END    JMP  XITVBV     ;Exit the VBI
0666        0590        .END
```




THAT'S NOT WHAT I MEANT WHEN I TOLD YOU TO DUMP THE PROGRAM.
I'LL GET THE BROOM.

BLOCK MOVES USING STRINGS

by R. S. Vickery

ATARI BASIC is great for handling strings of characters. There is a way to define the entire memory space of the computer as one gigantic string and to move blocks of data anywhere in memory with simple string commands. If we call this string A$, then ASC(A$(104,104)) is the same as PEEK(104) and A$(32768,32768+127)=A$(57344,57344+127) is equivalent to, and a lot faster than, FOR I=0 to 127:POKE 32768+I,PEEK(57344+I):NEXT I. This procedure can be used to copy the ROM character set into RAM (as in the sample program) or to move players vertically.

Notes on the Program:

Line 10: A$ is defined before any other strings, arrays or variables so that its data will be at the start of the Variable Value Table.

Line 20: The address of the start of the Variable Value Table is stored in the variable VVTP.

Line 30: The address of the start of the string/array storage area is stored in the variable STARP.

Lines 40-70: BASIC is tricked into thinking that A$ extends from address 1 to address 65535. The Variable Value Table contains eight bytes of data for each string, array or variable. After line 70 has executed, the first eight bytes in the table will look like this:

ADDRESS	VVTP	VVTP+1	VVTP+2,VVTP+3	VVTP+4,VVTP+5	VVTP+6,VVTP+7
VALUE	129	0	1-STARP	65535	65535

The fist byte is a code indicating that A$ is a dimensioned string. The second byte is the variable number of the string, 0 indicates that A$ was the first string, array or variable to be defined. The third and fourth bytes make up a 2-byte integer giving the offset of A$ from the start of the string/array storage area. When BASIC wants to find the address of A$, it adds this offset to the address of the start of the string/array storage area and gets 1 as the answer (i.e. 1-STARP+STARP=1). The next two bytes give the current length of the string [you'll be more familiar with this as the value returned by the LEN() function - GF] and the last two bytes give the dimensioned length.

Line 80: RAMTOP is moved down to make room for the character set.

Lines 90-110: The character set is copied from ROM to RAM. The copying is done in blocks of 128 characters because of a bug in BASIC. [If the string being copied is a multiple of 256 bytes, the correct number of bytes are copied, but they are copied to a place 256 bytes beyond where they should go - GF]

```
1 REM #############################
2 REM #   BLOCK MOVE USING STRING   #
3 REM #       by R. S. Vickery      #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)     #
6 REM #          June 1983          #
7 REM #############################
10 DIM A$(1):RAMTOP=106:CHBAS=756
20 VVTP=PEEK(134)+256*PEEK(135)
30 STARP=PEEK(140)+256*PEEK(141)
40 OFFSET=65537-STARP:HI=INT(OFFSET/25
6):LO=OFFSET-256*HI
50 POKE VVTP+2,LO:POKE VVTP+3,HI
60 POKE VVTP+4,255:POKE VVTP+5,255
70 POKE VVTP+6,255:POKE VVTP+7,255
80 CRAM=PEEK(RAMTOP)-4:POKE RAMTOP,CRA
M-4:GRAPHICS 0
90 CRAM=256*CRAM:CROM=57344
100 FOR I=0 TO 896 STEP 128:A$(CRAM+I,
CRAM+I+127)=A$(CROM+I,CROM+I+127):NEXT
 I
110 POKE CHBAS,CRAM/256
120 END :REM Your program starts here.
```

MORE ON FINE SCROLLING

by Jon Atack

[Reprinted from the Atari Computer Enthusiasts Newsletter, February 1983. Subscription enquiries to A.C.E., 3662 Vine Maple Drive, Eugene, OR 97405, U.S.A.]

In the December/January 1983 issue of the Eugene A.C.E. Newsletter, we covered Vertical Blank Interrupts. In this column, we'll be using VBI's together with vertical and horizontal fine scrolling.

To achieve fine scrolling, there are basically three things you must do:

1. Make a custom display list.
2. Set up a background to be scrolled.
3. Write a VBI routine to do the scrolling.

A display list is a set of instructions telling the ANTIC chip what part of memory is to be displayed and how to display it. For our purposes, we'll use a display list to scroll the whole screen both vertically and horizontally.

HEX	DECIMAL	COMMENTS
70	112	8 blank lines
70	112	8 blank lines
70	112	8 blank lines
47	71	Load Memory Scan (LMS)
20	32	Tells ANTIC that screen
3C	60	memory starts at $3C20
37	55	ANTIC mode 7 (GR.2)
37	55	bits 4 & 5 set to enable
37	55	fine horizontal &
37	55	vertical scrolling
37	55	
37	55	
37	55	
37	55	
37	55	
37	55	
37	55	
41	65	Jump & wait for Vertical Blank (JVB)
00	0	Tells ANTIC that display list
30	48	starts at $3000

For most games, the background will be one of the special character set modes (ANTIC modes 4, 5, 6 or 7) used with a custom character set. Examples of games using this technique are Eastern Front, Protector, Wizard of Wor, Pac-Man, Centipede, Nautilus, Canyon Climber, Miner 2049er, etc. Almost every commercial game uses a custom character set background. However, to save time and space, our fine scrolling demo will just scroll over the data in the Operating System. To make your own background, you must

1. define your character set, then
2. place your screen data into memory someplace, and
3. change the Load Memory Scan pointer in the display list to point to your screen data.

Custom displays take a lot of time, but the results are well worth it. A good background is a must for any commercial game.

The final step is to write a routine to do the scrolling. There are two fine scroll registers in the Operating System called HSCROL and VSCROL. HSCROL controls horizontal scrolling and VSCROL controls vertical scrolling. To use them, simply poke or store the number of lines you want scrolled.

There's just one problem. Fine scrolling can only scroll so far. The vertical limit is 16 TV

GETTING STARTED WITH TEAM ATARI fig-FORTH

by John Mattes

1. Making a Working Copy

   1.1 Boot the Team ATARI Disk.

   1.2 After "fig-FORTH 1.4S" appears, type "30 LOAD". (Note that all spaces are relevant and must be included). This will load screens 31, 36, 37, 39 and 10. Some of these screens continue to other screens so it takes a while. The Screen Index provided on the disk (inspect by typing "0 LIST" stop scrolling with CTRL-1) says Screen 30 is a "Full Load". This is not so. A copy of the screen index is attached.

   1.3 After the "ok" appears at the end of "30 LOAD", type "7 LOAD". This will load the System Setup/Bootmaker verbs.

   1.4 Type "MAKEBOOT". You will be prompted to insert a formatted disk. Do so. DO NOT leave your Team ATARI disk in the drive or you will overwrite it.

   1.5 Follow the prompt by typing "Y" after the formatted disk is inserted. A self-booting copy of all of the verbs loaded in steps 1.2 and 1.3 will be written to your working disk. These will be automatically loaded into your system every time you boot the working disk. No more "30 LOAD" etc. If your disk was newly formatted, all screens will be full of zeros (displayed as hearts).

   1.6 Put your Team ATARI disk in a safe place.

2. Entering a Program

   2.1 Boot your working disk.

   2.2 Type "n EDIT" where n is a screen number of your choice. (Remember that the system leaves you in Hex). Try "F EDIT".

   2.3 Type "UL" to list the upper half of the screen. You'll get a screen-full of hearts.

   2.4 Type "F WIPE" to clear screen F of hearts.

   2.5 Using the cursor control keys, position the cursor somewhere up in the edit window and type in your program - try the one attached. Note that the RETURN key does not work. The TAB key is handy for returning quickly to the beginning of a line.

   2.6 Delete the ;S at the end of the top line using the cursor controls and delete/backspace or space bar. Unless this is done, the ;S will tell the compiler to cease interpreting your screen before your program is loaded. Put a ;S at the end of your program or the system will bomb when you try to load.

   2.7 After you've finished typing your program, move the cursor down to the line which reads "DOIT" and hit "RETURN". This loads the information on the screen into the screen buffer in RAM.

2.8 Type "FLUSH". This writes all updated screens (only screen F in this case) to disk.

2.9 Re-boot and type "F LIST". This should list your edited screen from disk. If you don't re-boot it will be listed from RAM.

3. Running Your Program

3.1 Type "F LOAD" to load screen F.

3.2 If there are no error messages, a simple "ok" will appear.

3.3 Now run your program by typing in the verb you want to run, e.g."WACKO".

TEST PROGRAM

```
( TEST PROGRAM JEM 14 APR 83 )

: WACKO 10 0 DO
        CR ." WACKO! I'VE DONE IT! "
        LOOP ;
;S
```

SCREEN INDEX

Screens	Contents
00/00	Screen Index
01/01	Source Credits
02/06	Error Messages
07/09	System Setup/Bootmaker
0A/0A	Single Drive Backup/Done
0B/0B	RPM Checker
0C/0E	
0F/0F	Decomp Disassembler Data
10/10	Editor Loader
11/14	Screen Editor
15/17	Line Editor
18/1A	Additional Editor Words
1B/22	Decompiler/Stack Display
23/24	Copies/Duplicate
25/26	Editor Expansion
27/27	1.4S Kernel Mods
28/2E	Assembler
2F/2F	DECUS Mods
30/30	Full Load
31/35	Hardware/Graphics/Sound
36/36	PON.POFF
37/38	RS232 Support
39/39	Display List Stuff
3A/3B	Player/Missile and Old Clone
3C/3D	Expansion
3E/41	Lineprinter Words
42/45	Formatted List Program

Odds & Sods

A regular column by Garry Francis

NEW PRODUCTS

Here is a list of software that is expected to be released by ATARI within the next 5 or 6 months. They are in approximate chronological order. Prices will not be known until they are actually released. Note the number of 16k cartridges! Family Finances (D), Dig Dug (16k ROM), E.T. Phone Home (16k ROM), Mickey in the Great Outdoors (D or C), Paint (D), Revised ATARI BASIC (8k ROM), Donkey Kong (16k ROM), Eastern Front (16k ROM), Soccer (16k ROM), Tennis (16k ROM), Mickey/Peter Pan (D or C), Donkey Kong Jr. (16k ROM), Ms. Pacman (16k ROM), Pengo (16k ROM), Pole Position (16k ROM), ATARI LOGO (16k ROM), (American) Football (16k ROM), Superman III (16k ROM), Joust (16k ROM), Robotron (16k ROM), ATARI Music I (D or C) and ATARI Music II (D or C). I'm itching to get my hands on Dig Dug, Pengo and Robotron!

* * *

Page 6 is a new ATARI magazine that is being published bi-monthly by ABACUS in England. The Editor was good enough to send us a copy of Issue 2 February/March 1983. It is 28 pages of interesting news and articles, including several program listings. There's a game called Four in a Row, a routine for drawing cards in card games, Tiny Text, a lot of stuff on GTIA and several demos. Australian subscriptions are 7 pounds for surface mail and 15 pounds for air mail. Cheques should be made payable to ABACUS and sent to Page 6, 18 Underwood Close, Parkside, Stafford, England.

* * *

Want a dust cover for your computer? Coverware, P.O. Box 719, Chatswood, N.S.W., 2067 (Phone (02) 498-5631) have some really top notch, custom-made dust covers for the ATARI 400 and 800. They're made from anti-static, cotton-backed vinyl and come in the same colour as the plastic case on the computer. Val Javen of Coverware sent us a sample, which I've been "testing" on my 800 and I'm impressed. It's a very snug fit with provision for power cables and connectors (but not for joysticks). The dust covers are available direct from Covercare for $15 including postage. This is a 25% discount on the normal price especially for A.C.E. members. Orders must be accompanied by cheque or money order and make sure you specify whether it's for a 400 or 800.

* * *

Flintech Computer Systems, P.O. Box 450, Nelson Bay, N.S.W., 2315 has a full-stroke keyboard for the 400. They are "...very simple to install (and remove if required). No modification of the ATARI 400 is required. Each keyboard is mounted in a sturdy plastic case with metal base and is supplied with complete instructions, cable and connector." They're priced at $90 each ($80 if you buy 10 or more). Flintech are also working on a 256k memory system for the 400 which "...will cost under $400 and will be available by the middle of this year." Write to the above address for further info.

* * *

One very unusual item turned up in our mail recently. It was a copy of an 8 page monthly publication called Farm Computer News. Its content should be obvious from the title. If any of our members are farmers and would like further info, or if you've got news of ATARI's in rural applications, contact Peter Cross at Rural Press Limited, P.O. Box 238, Edgecliff, N.S.W., 2027 Phone (02) 326-1144.

* * *

CLUB NEWS

Paul Phillips was forced to resign from his position as Bulletin Board "investigator" due to personal committments. Roberto Romano has stepped in to fill his place. He will continue to be assisted by Vince Grogan. For those not aware of it already, we are doing a feasibility study on the setting up of a Bulletin Board Service. There are a multitude of obstacles to overcome before we can get started, but if it works out, it will be the first ATARI Bulletin Board Service in Australia. Please contact Roberto on (02) 327-4898 if you can help out in any way, no matter how small. We particularly need the hardware and information.

* * *

At the May meeting, we reintroduced short talks like we used to have in the old days. Tony Reeve talked about the components that make up the ATARI, John Mattes and John Massara gave an introduction to FORTH using the Team ATARI FORTH from the Software Exchange for the examples and Peter Bamford showed us how to find the typos in a program that you've typed in from a magazine. The format was very popular, so we'll continue with it in future. We hope to have 2 or 3 short talks of 15-30 minutes each. In this way, there is plenty of variety, people won't get bored and there is still plenty of time for software demos, general discussion and getting to know each other. Giving a talk is much easier than it seems. You needn't worry about looking like a fool because we're a small, close-knit club and everyone's there to enjoy themselves. So give it a go! If you've got a favourite topic that you could talk about (or perhaps you'd like to show off your latest programming masterpiece) then contact Peter Bamford (043) 42-2655 (Home), (043) 40-1201 (Work) or Peter Dickeson (02) 411-1758 (Home), (02) 2-0230 Ext. 5133 (Work).

* * *

At the May meeting, we also reintroduced a suggestion box. Here are some of the suggestions received. If you can help out with any of them, ring Tony Reeve (02) 452-2974 (Home) or (02) 923-9586 (Work).

1. Could we have a talk on DLI's?
2. Could we have something on ANTIC?
3. Could we organise 2 or 3 days of instruction on the ATARI computer (i.e. use of, graphics, problem solving, use of hardware, compatability of other brands of hardware)?
4. Get a better suggestion box!

* * *

We will be having raffles at some of the future meetings to (hopefully) raise some money. Prizes will generally be software - starting with some of the excellent Thorn EMI range. If you've got anything that would be suitable as a prize and you can donate it to the club, please contact Barry Williams (02) 452-2229 (Home) or (02) 359-2424 (Work).

* * *

OTHER NEWS

Colin Macpherson (Education Consultant for Futuretronics) has been able to supply some interesting details of what's being done with the ATARI in Australia. I'll let him explain.

> "Regarding novel projects that are using ATARI computers, I am sure you are aware of more developments than myself. However, I do know that a doctor at one of the major hospitals is using an ATARI computer to monitor bacterial growth in AGAR solutions and that an ATARI figures prominently on the set of a new ABC children's television series called Infinity Limited. Perhaps the most interesting project presently being undertaken that utilises an ATARI computer is Karl Lind's search for intelligent signals from the neighbourhood of the star Sigma Draconis. I should add that this research is being pursued in the U.S. not Australia (for further details see Newsweek 31st January, 1983)."

I have heard rumours that the ATARI was used for some backdrops or something in the new TV series Knight Rider. And also that a forthcoming Australian TV series features an ATARI as an aid in the rehabilitation of a man who is handicapped by a serious injury.

* * *

Colin was also able to give me some information about computer courses being held at the Royal Melbourne Institute of Technology. There were 3 Holiday Computer Courses for Beginners last January using ATARI computers supplied by Futuretronics. The cost was $90. There's apparently a new course in May (9th-13th?) for $120. The details are a bit vague, but from what I can make out, this is an advanced course run by David Jones-Ellis, the publisher of Microprobe. Sounds interesting. Why can't we have a course in Sydney? Who's interested? If there was a serious course available (and we could get computers for hands on work), how much would you be willing to pay?

* * *

Futuretronics also donated $25,000 worth of computer equipment to R.M.I.T. for their ATARI Mobile Computer Laboratory. This contains 10 computers and necessary support hardware, such as disc drives and printers. It will be transported to various technical and high schools for student education in the day and computer awareness programs for "oldies" in the evenings. Half their luck! They never had anything like that when I was at school.

* * *

The Australian Computer Education Conference was held at La Trobe University in Melbourne from 16th-18th May, 1983. I wish I'd known about it earlier, as it sounded like an exciting event. There were sessions on every conceivable aspect of computers in education and speakers included representatives from ATARI, Inc., Minnesota Educational Computing Consortium (who receive a lot of assistance from ATARI) and Futuretronics. An exhibition ran during the conference and Futuretronics was there with our beloved ATARI.

* * *

Our friends in Queensland will be interested to know that The Queensland ATARI Resources Club meet at 7.30 P.M. on the 3rd Wednesday of each month in Seminar Room S17, Lower Level, Bardon Professional Development Centre, 390 Simpsons Road, Bardon. If interested in joining or helping the group out, contact Ken Morcom, P.O. Box 48, Clayfield, Qld, 4011 or ring Dirk Bakker on 390-2906. To the best of my knowledge, there must now be an ATARI users' group in nearly every state and territory in Australia. Let us know of any that we haven't previously mentioned in INSIDE INFO.

* * *

The price of the ATARI 800 in the United Kingdom recently dropped to 399.99 pounds. In Australian currency, that's about $730. In the U.S., they practically give 'em away! How come we still have to pay outrageous prices? Does anyone know?

* * *

ADVENTURE S.I.G.

The Adventure S.I.G. has been fairly withdrawn until recently. A few weeks back, we had an "Adventure Weekend" at Peter Bamford's place at Umina and we had a ball! We solved the SoftSide Adventure "Arrow One", which had been giving me trouble. (You see, there was this communication problem with the librarian. Whenever I tried to get the encyclopedia, she'd say "DVPL RUMBAT?". I thought the frotti would help, but it turned out to be for something else. Eventually, I gave her away and confronted the alien dog to steal the helki from the garden. When I gave these to the Xippi Bird, it rewarded me with some Xippi eggs for my generosity. I was able to feed these to the baby Gooba, who rapidly grew to adult size and carried me across the sea to the City of Pfitt. When I found The Professor, he gave me a means to return to the City of Renn, but upon returning, I couldn't find his daughter or use the teleporter, as it was always being recharged.)

We also tackled Datasoft's brilliant new graphics Adventure called "The Sands of Egypt". This Adventure has all the intrigue of a Scott Adams-type text Adventure, but with the addition of brilliant high resolution, full colour graphics and some animation. Unfortunately, we couldn't solve it. I suspect there's a way to enter the pyramid, but we couldn't find it. Can any of our overseas readers help?

On the second day, we attempted Deadline. This game is fantastic. For those not familiar with it, you have to solve the murder of Marshall Robner, a well-to-do industrialist. We spent about 5 absorbing hours exploring the Robner mansion, questioning suspects and looking for clues. In the excitement, we missed the reading of the will, but we've found several good leads. Even so, we're no nearer to solving the murder than when we first started.

We also spent quite a bit of time playing arcade games, like Protector II (absolutely aweful) and Fort Apocalypse (one of the best ATARI arcade games I've played to date). We'll have another "Adventure Weekend" some time in the near future.

A number of people have asked about joining the Adventure S.I.G., so here's some details. Simply tell Peter Bamford that you're interested and give him your name, address and 'phone number, together with a written list of all the adventures you've completed. An Adventure is not considered complete until you've discovered every location, drawn a map, explored all the sub-plots and achieved the ultimate goal. In other words, if someone asks you for a hint, you should be able to give it to them without any trouble. We'd particularly like original Adventures and translations from other machines. Beginning Adventurers are also welcome. Peter may be contacted direct at 4 McManus Close, Umina, N.S.W., 2257. Phone (043) 422655 (Home) or (043) 401201 (Work).

* * *

FORTH S.I.G.

John Mattes has agreed to co-ordinate the FORTH S.I.G. John is a real whiz-kid at FORTH. He has already written some documentation for the Team ATARI FORTH disc (available through the Software Exchange for $10) and will be working on a glossary of its extra words. He has also promised an update to "Escape" (Issue No.6) so that it'll run on this FORTH. If you are interested in FORTH, have suggestions as to what the S.I.G. should do or can help in any way, please contact John on (02) 94-5463 (Home) or (02) 230-5710 (Work).

* * *

QUESTIONS AND ANSWERS

Is it possible to copy a self-booting cassette and if so how?

- Arthur Banks

You didn't say whether you wanted to copy it to another cassette or to disc, but the following information should help. A cassette boot file contains a 6 byte header with information required by the system to load and execute the file. Of these 6 bytes, the 1st byte is not used. The 2nd byte contains the number of 128 byte records to be read in as part of the boot process. The 3rd and 4th bytes contain the address to which the program will begin to be loaded. This is in normal 6502 format, i.e. LSB followed by MSB. The 5th and 6th bytes contain the address to which control will be transferred after the boot is complete. (Full details and examples are given in the Operating System Users' Manual, specifically pages 148-154.) With this information, you should find it an easy task to write a program to back-up your master cassette. You can then put the master cassette in a safe place and use the back-up for normal usage. But don't give copies of commercial programs to your friends! This is theft and/or infringement of Copyright and is punishable by law. For those who don't yet have the necessary programming skills, you could use "Cassette Boot Manager" from ANTIC February/March 1983. This will copy to cassette or disc. There's also a program in COMPUTE! November 1982, but this will copy to disc only. I don't think either of these will copy multiple boot programs.

* * *

Do you know how to stop the automatic scrolling in GRAPHICS 0?

- R. S. Vickery

I'm not sure that I know what you mean. Pressing BREAK or SYSTEM RESET will stop it good and proper, but I somehow don't think that's quite what you want. Remember that you can always start and stop the scrolling (or any printing to the screen) at your leisure using CTRL 1. (See BASIC Reference Manual page 14.) This is particularly useful when listing a program. I hope this is the answer you were looking for.

* * *

HINTS AND TIPS

It has recently come to my attention that every man and his dog believes that page 6 is reserved solely for him to use as he pleases. This is the greatest fallacy since they thought the world was flat! Page 6 is not reserved and never has been! Let me explain. The Operating System Source Listing clearly states that page 6 (as well as page 5 and the upper half of page 0) is reserved as a "user work space". This means that any application outside the Operating System itself can use this area. BASIC is such an application! The BASIC Reference Manual said that page 6 was "Free RAM" and BASIC never "seemed" to use page 6, so everybody started to use it. However, "free RAM" is not the same as "reserved RAM" and BASIC does use this area. This happens whenever an INPUT statement attempts to input more than 128 characters. In this case, the bottom half of page 6 is used as a buffer. This will only happen when the input is from a disc or cassette, as an input from the screen is restricted to 1 logical line - a maximum of 120 characters. It turns out that ATARI BASIC is not the only culprit. I believe Microsoft BASIC uses all of page 6 and BASIC A+ uses 1552-1602 [$610-$642]. It is quite likely that other languages also use page 6. The moral is "don't use page 6 unless it is within a controlled environment". By this, I mean that you can use it for a game or other program where the memory configuration is under your control and you know that you will not be using INPUT from an external device. Do not use page 6 for system utilities or subroutines, device handlers and so on. The flashing cursor program from Issue No.4 is a prime example of when not to use page 6. An input of over 128 characters will change the Vertical Blank Interrupt service routine and the system will almost definitely crash as a consequence. I hope this solves some people's page 6 problems.

* * *

Remember that DOS 2 will only address 2 drives unless you specifically POKE it for more. This is explained in Appendix G of the Disk Operating System II Reference Manual, but it's amazing how many people don't bother to read their manuals. To make the change, begin by booting DOS 2. Memory location 1802 can be changed to tell the system which drives will be used. This is done by setting any or all of bits 0-3 to correspond to drives 1-4 respectively. If a bit is set, a buffer will be reserved for the appropriate drive. For example, use POKE 1802,7 to use drives 1-3 or POKE 1802,15 to use all 4 drives. You can also use POKE 1802,1 if you've only got 1 drive. This saves some memory as the buffer for drive 2 is not set up (the default is POKE 1802,3 for drives 1 and 2). It is also legal to use say POKE 1802,5 to use drives 1 and 3. To make the changes permanent, type DOS, then type H to write the modified DOS files to disc. Never send any programs to the Editor or the Software Exchange with modified DOS files.

* * *

Here is the "neat trick of the month". Have you ever wanted to get an ASCII dump of a program (perhaps to see the program in its tokenised form)? Try typing SAVE "S:". (You could also use SAVE "E:", but the cursor control characters will be executed, making things somewhat difficult to read.)

* * *

When writing a program recently, I had to do some research on maskable interrupts (IRQ) and quickly realised that I had previously been disabling the BREAK key with a wrong POKE. Although this POKE still works, I thought I'd better correct the situation, as several programs have already copied my incorrect procedure. The correct POKEs for enabling or disabling the BREAK key and/or keyboard are as follows:

Enable BREAK key, enable keyboard: POKE 16,192:POKE 53774,192
Enable BREAK key, disable keyboard: POKE 16,128:POKE 53774,128
Disable BREAK key, enable keyboard: POKE 16,64:POKE 53774,64
Disable BREAK key, disable keyboard: POKE 16,0:POKE53774,0

You must store the desired value in both the shadow register POKMSK at address 16 [$10] and the write only hardware register IRQEN at address 53774 [$D20E]. Note that the above POKEs do not take the other 6 IRQ interrupts into account. Refer to the Hardware Manual pages II.29, III.2 and III.3 for details of these and a very good (though confusing) explanation. The following programs from past issues should have the POKE 53774,119 changed to POKE 53774,64...GRAPHICS 0 Colour Selector and ANTIC Mode 4 Demo from Issue No.2, Organ from Issue No.3, Pacman's Revenge, Chessboard and Doggies from Issue No.5.

* * *

When a 2-byte number is stored in memory in the standard format of Least Significant Byte followed by Most Significant Byte, it may be converted to a single decimal number for storage in a variable as follows:

```
NUM=PEEK(ADR)+256*PEEK(ADR+1)
```

where NUM is the number you want and ADR is its address in memory. The reverse process is equally simple:

```
HI=INT(NUM/256):LO=NUM-256*HI
POKE ADR,LO:POKE ADR+1,HI
```

* * *

FOR SALE

If interested in any of the following items, ring Tony Reeve (02) 452-2974 (Home) or (02) 923-9586 (Work).

1. ATARI 810 Disc Drive. Brand new. Never used. $700
2. COMPUTE!'s Second Book of ATARI. $10
3. ATARI BASIC language plus full documentation. $50

* * *

PRODUCT UPDATE
ATARI HOME COMPUTER SYSTEM

ATARI® BASIC Reference Manual Update

This product update contains a number of corrections and additions to the *ATARI BASIC Reference Manual.*

Page 1. This definition is missing from the **TERMINOLOGY** list:

Floating Point Number: A number containing an integer part, a decimal point, and a fractional part. The total number of significant digits in a floating point number, excluding the exponent, may be either nine or ten. This depends on whether the exponent is an even or odd multiple of 10.

Page 6. This information pertains to the **ARITHMETIC OPERATORS** subtraction and exponentiation:

Note: Avoid negating zero, as this will produce an invalid number. For example, if you type

```
PRINT -0
```

the result will be

```
-0E-<8
```

Note: Since the algorithm used to generate exponents (^) is only an approximation, you cannot obtain integer results with it—for example, 2^2=3. 99999996. To correct this, use the following technique:

```
X=2^2
PRINT INT (X+.5)
4
```

Page 7. This Note regards the use of the **LOGICAL OPERATORS:**

Note: Avoid using the statement PRINT A=NOT B, as the results are not predictable. Essentially, any PRINT statement with a NOT operator will be unpredictable.

Page 13. This Note is in reference to **SCREEN EDITING:**

Note: Large amounts of editing may lock up the system. It's recommended that programs under development be stored to cassette or diskette periodically (every 30 or 40 edits) with the SAVE or CSAVE command.

Page 20. This Note regards **ON/GOSUB** statements:

Note: If an ON/GOSUB expression evaluates to a number greater than the number of subroutine entries, then a POP statement will be necessary to clear the stack (see POP command, Section 4).

Page 22. Further information on **RESTORE (RES.):**

The RESTORE statement will not generate an error if the line number referenced does not exist. Instead it will RESTORE to the next larger line number in the program. Care should be taken to update RESTORE statements when renumbering a BASIC program.

Page 25. Some additional information on using the **INPUT (I.)** statement:

When executing an INPUT from the screen, avoid moving the cursor away from and then back to the same line; otherwise, the wrong data may be input. Specifically, the INPUT prompt will be included in the INPUT string.

If a string of 128-255 characters is INPUT, then RAM locations 1536-1664 will be overwritten. This area is normally reserved for storage of programs or data. (See the *ATARI Tech Reference Notes.*)

To INPUT strings of more than 127 characters, use the GET command and store the values into a string (see Section 5, OPEN/CLOSE and PUT/GET commands).

Note: The maximum number of characters that can be INPUT from the screen is 120. The maximum for other devices is 255.

Note: Make sure that every INPUT statement has a variable after it; otherwise, unpredictable results may occur.

Page 26. This regards the use of the **LOAD (LO.)** command:

Note: If a program is loaded that is too large for the available memory space, it may give unpredictable results without an error message.

This Note should follow the **LPRINT (LP.)** command description:

Note: An LPRINT command with a semicolon at the end will cause the following LPRINT statement to print on the next 40-column tab. A 40-column printer will move to the next line in such a case. To use the semicolon effectively, use the OPEN statement for the printer, then write to the printer with a PRINT statement (see OPEN/CLOSE and PRINT commands, Section 5).

Page 27. This information pertains to the filespec definition:

Note: Be sure to include the closing quotation marks on a filespec parameter, especially when putting multiple statements on one line. For example,

```
OPEN #1, 4, 0, "D:TEST":STOP
```

will work, but

```
OPEN #1, 4, 0, "D:TEST:STOP
```

will not function correctly.

Page 28. This is an addition to the **POINT (P.)** section:

Note: To update a file, you must open it with a 12 in aexp1.

In the last paragraph under **PRINT (PR. or ?)**, the first sentence should read:

A comma tabs every 10 spaces.

The following sentence should conclude the final paragraph on **PRINT (PR. or ?)**:

However, if the last character to be printed (as in a string with quotation marks) is a CTRL R or CTRL U, then the next PRINT will begin at the end of the current line.

This note should then conclude this section on **PRINT (PR. or ?)**:

Note: In rare circumstances data printed to a diskette may have part of the BASIC program embedded in it. If this occurs, retry the operation.

This Note regards the **PUT (PU.)/GET (GE.)** section:

Note: In certain circumstances the GET function may modify other variables within the program. To avoid this, PRINT any number to the screen between each GET.

Page 30. Here is a corrected version of the table—note in particular the correction on cmdno 32:

cmdno	OPERATION	EXAMPLE
3	OPEN	Same as BASIC OPEN
12	CLOSE	Same as BASIC CLOSE
13	STATUS REQUEST	Same as BASIC STATUS
17	DRAW LINE	Same as BASIC DRAWTO
18	FILL	See Section 9
32	RENAME	XIO 32,#1,0,0,"D:TEMP,CAROL"
33	DELETE	XIO 33,#1,0,0,"D:TEMP.BAS"
35	LOCK FILE	XIO 35,#1,0,0,"D:TEMP.BAS"
36	UNLOCK FILE	XIO 36,#1,0,0,"D:TEMP.BAS"
37	POINT	XIO 37,#1,A,B
38	NOTE	XIO 38,#1,X,Y
254	FORMAT	XIO 254,#1,0,0,"D2:"

Page 33. The last sentence in the paragraph about the **CLOG** function should read:	CLOG(0) through CLOG(1) are inaccurate and should not be used.
Page 34. The last sentence in the paragraph about the **LOG** function should read:	LOG(0) through LOG(1) are inaccurate and should not be used.
Page 38. The last line in the first paragraph should read:	was stored there previously.
Page 39. The first sentence should read:	Upon execution, the screen displays THE SQUARE ROOT OF 10000 IS 100.
In the second paragraph, the last line should read:	number 1000000000.
This is additional information on the **VAL** function:	Only the numeric field will be translated, while the text will be ignored. For example: VAL("5SUM")=5
This information pertains to **String Concatenation:**	**Note:** BASIC cannot move strings of 256-character multiples correctly. String lengths should be checked; if any string contains a multiple of 256 characters, add or subtract one character from the amount to be moved.
In *Figure 7-6*, the correct result of the program on the left is:	BCD#
Page 42. Some additional information on using the **DIM (DI.)** statement:	Make sure that the DIM statement does not contain a space between the string or array name and the left parenthesis of the dimensioned amount; otherwise, the following will happen— DIM L (10) becomes DIM L10) —and this variable can no longer be referenced. **Note:** The command COM is identical to DIM and may be used in its place.
Page 43. This is an additional Note for the **DIM (DI.)** section:	**Note:** Due to a discrepancy in boundary checking, arrays of up to 32766 by 32766 in size can be dimensioned. The programmer should size the array ahead of time to ensure that there is no "virtual" storage space.
Additional information on using the **CLR** command:	The second sentence in the last paragraph, beginning "It also clears ...," should be deleted. The CLR command will not initialize the values in strings and arrays.

Page 45. Here is a corrected version of **TABLE 9.1:**

TABLE 9.1—TABLE OF MODES AND SCREEN FORMATS

Gr. Mode	Mode Type	SCREEN FORMAT Horiz. (Columns)	Vert. (Rows) Split Screen	Vert. (Rows) Full Screen	Number Of Color Registers	Split Screen	RAM Required Bytes) Full Screen
0	TEXT	40	—	24	1½	—	992
1	TEXT	20	20	24	5	674	672
2	TEXT	20	10	12	5	424	420
3	GRAPHICS	40	20	24	4	434	432
4	GRAPHICS	80	40	48	2	694	696
5	GRAPHICS	80	40	48	4	1174	1176
6	GRAPHICS	160	80	96	2	2174	2184
7	GRAPHICS	160	80	96	4	4190	4200
8	GRAPHICS	320	160	192	1½	8112	8138

Page 49. The last sentence under **PLOT (PL.)** should read:

"The range of points begins at 0 and extends...."

Page 50.

In **TABLE 9.3**, the color PURPLE should be inserted after PINK in the first column, and the number 5 should be inserted after 4 in the second column.

Page 51. The sentence directly under **TABLE 9.4** should read:

"DEFAULT" occurs if no SETCOLOR statement is used.

Page 53. Here is a corrected version of **TABLE 9.5:**

MODE, SETCOLOR, COLOR TABLE

Default Colors	Mode or Condition	SETCOLOR (aexp1) Color Register No.	Color (aexp)	DESCRIPTION AND COMMENTS
	Mode 0 and all text windows	0	Color data actually determines character to be printed	—
LIGHT BLUE		1		Character luminance (same color as background)
DARK BLUE		2		Background
		3		—
BLACK		4		Border
ORANGE	Modes 1 and 2 (text modes)	0	Color data actually determines character to be printed	Character
LIGHT GREEN		1		Character
DARK BLUE		2		Character
RED		3		Character
BLACK		4		Background, border
ORANGE	Modes 3, 5, and 7 (four-color modes)	0	1	Graphics point
LIGHT GREEN		1	2	Graphics point
DARK BLUE		2	3	Graphics point
		—	—	—
BLACK		4	0	Graphics point (background default), border
ORANGE	Modes 4 and 6 (two-color modes)	0	1	Graphics point
		—	—	—
		—	—	—
		—	—	—
BLACK		4	0	Graphics point (background default), border
	Mode 8 (1 color, 2 luminances)	—	—	—
LIGHT BLUE		1	1	Graphics point luminance (same color as background
DARK BLUE		2	0	Graphics point (background default)
		—	—	
BLACK		4	—	Border

Page 54. In *Figure 9-4,* line 80 should read:

```
80 XIO 18, #6, 12, 0, "S:"
```

Page 55. This information pertains to **TABLE 9.6:**

In Column 1, # 14, a period, not a bar, shows on the screen.

In Column 3, #'s 92-95 should show a superscripted circled 1 next to their characters.

Page 56. Here is a corrected version of **TABLE 9.7:**

TABLE 9.7—CHARACTER/COLOR ASSIGNMENT

		Column 1 Conversion	Column 2 Conversion	Column 3 Conversion	Column 4 Conversion
MODE 0	[2]SETCOLOR 2	#+32	#+32	#-64	NONE
		POKE 756,224		POKE 756,226	
MODE 1 OR MODE 2	SETCOLOR 0	#+32	#+32	#-32	#-32
	SETCOLOR 1	NONE	#+64	#-64	NONE
	SETCOLOR 2	#+160	#+160	#+96	#-96
	SETCOLOR 3	#+128	#+192	#+64	#-128

[2] Luminance controlled by SETCOLOR 1, 0, LUM

Page 58. The last paragraph should read as follows:

Note that the DATA statement in line 90 ends with 256, which is outside of the designated range. The 256 is.....

In **TABLE 10.1:**

The PITCH VALUE of 193 should have a musical note of "E," not "D."

Page 63. The last line in item 9 should read:

precedence will save a few bytes.

Page 67. In *Figure 11-2,* line 0260 under Data should be:

#2

Page E-1.

The right parentheses are missing after the word "CONSTANT" in Atari Functions of Inverse Cosine, Inverse Secant, and Inverse Cosecant.

Page H-7. Line 160 in the program should read:

```
160 IF K=125 OR K=155 THEN 180
```

Page H-8. Line 50 in the program should read:

```
50 PLOT 0,0:DRAWTO 159, DR
```

Page 117.

Following COM, "(see DIM)" should be deleted and replaced with "A-1."

Page 118.

Under "Input/Output Devices," Line Printer should be followed by "(P:)," not "(L:)."

Page 119.

"NOTE, 26" is missing from the listing.

Page 120.

"STATUS, 29" is missing from the sublisting under "Statement" and also from the regular listing.

INSIDE BACK COVER
here is the corrected table:

MODE, SETCOLOR, COLOR TABLE

Default Colors	Mode or Condition	SETCOLOR (aexp1) Color Register No.	Color (aexp)	DESCRIPTION AND COMMENTS
	Mode 0 and all text windows	0	Color data actually determines character to be printed.	—
LIGHT BLUE		1		Character luminance (same color as background)
DARK BLUE		2		Background
		3		—
BLACK		4		Border
ORANGE	Modes 1 and 2 (text modes)	0	Color data actually determines character to be printed.	Character
LIGHT GREEN		1		Character
DARK BLUE		2		Character
RED		3		Character
BLACK		4		Background, border
ORANGE	Modes 3, 5, and 7 (four-color modes)	0	1	Graphics point
LIGHT GREEN		1	2	Graphics point
DARK BLUE		2	3	Graphics point
		—	—	—
BLACK		4	0	Graphics point (background default), border
ORANGE	Modes 4 and 6 (two-color modes)	0	1	Graphics point
		—	—	—
		—	—	—
		—	—	—
BLACK		4	0	Graphics point (background default), border
	Mode 8 (1 color, 2 luminances)	—	—	—
LIGHT BLUE		1		Graphics point luminance (same color as background)
DARK BLUE		2	2	Graphics point (background default)
		—	—	
BLACK		4	—	Border

C061038 REV A
©1982 ATARI, INC

COMPUTER WAVE
PTY LIMITED

LOWER GROUND FLOOR - GRACE BROS
& ALSO 325 GEORGE ST - SYDNEY

Exiting new store opens 24th May 1983 in a prime location at 325 George Street

Between Wynyard Station and Martin Place

Open Saturdays 9-4pm will try to open Sundays 12-5pm

At last you will be able to see our huge range of software & hardware on display. Watch our windows for special offers.

PHONE 231 5791 235 2162